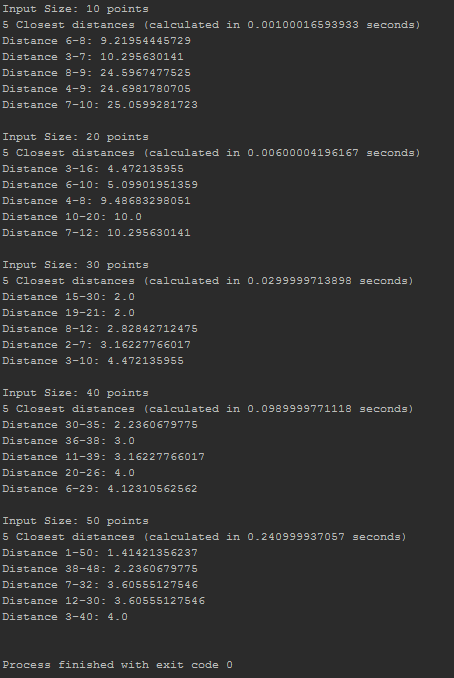
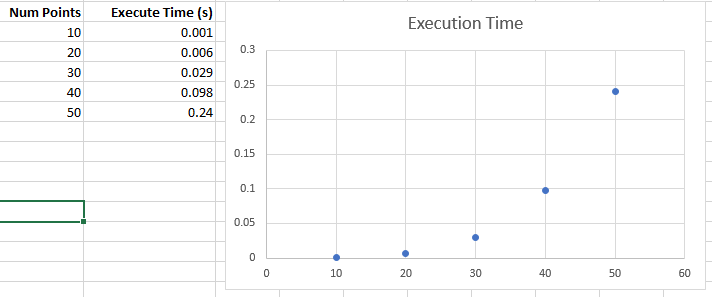
Closest Pairs

1. Pseudocode:  
   // This is the data structure we will return, contains both the // distance between 2 points identified in the name **DistancePair**  
    int distance  
    string pairName  
     
   // This method calculates the distance  
   **Distance(x1, y1, x2, y2)**  
    distance = (x2-x1)2 + (y2-y1)2  
    distance = sqrt(distance)  
    **return distance**  
     
   // This method produces the m closest pairs  
   **ClosestPairs(P, m)**  
    DistancePairs = new List()  
     
    // first calculate the distance between each 2-combination  
    **for i = 1 to P.length**  
    P1 = P[i]  
    **for j = (i+1) to P.length**  
    P2 = P[j]  
    distance = Distance(P1.x, P1.y, P2.x, P2.y)  
    pairname = P1.name + “-“ + P2.name  
    DistancePairs.add(new DistancePair(distance, pairname))  
    **end for**  
    **end for**  
     
    // use a simple bubble sort to sort DistancePairs  
    **for i = 1 to DistancePairs.length** **for j = 1 to (DistancePairs.length – i - 1)**  
    if DistancePairs[i].distance > DistancePairs[i+1].distance  
    swap(DistancePairs[i], DistancePairs[i+1])  
    end if  
    **end for**  
     
    **return DistancePairs.sublist[0..m]  
     
   Worst-Case running time:**  
     
   Since the distance calculation runs in constant time and calculating there will always be the same number of 2-combinations for a given set, the worst-case time complexity is that of the bubble sort used to arrange the set of distance pairs from smallest to largest – O(n2)
2. **The output of main.py below:**  
   Graphing the input size and execution times gives the following:  
     
   An exponential increase in time as the number of points increases – exactly the same as the algorithm’s worst-case running time
3. Because the performance of the above algorithm is tied to the sorting of the calculated distances the most straightforward way to reduce the worst-case performance of the algorithm would be to use a better sorting algorithm like merge sort or quick sort.